

DIVIDE

DIVIDE [**ROUNDED**] *operand1* **INTO** *operand2* [**GIVING** *operand3*]

Operand	Possible Structure				Possible Formats												Referencing Permitted	Dynamic Definition
Operand1	C	S	A	N		N	P	I	F								yes	no
Operand2	C	S	A	NM*		N	P	I	F								yes	no
Operand3		S	A		A	N	P	I	F	B							yes	yes

* "N" if GIVING clause is used, "M" if GIVING clause is not used.

DIVIDE *operand1* **INTO** *operand2* [**GIVING** *operand3*] **REMAINDER** *operand4*

Operand	Possible Structure				Possible Formats												Referencing Permitted	Dynamic Definition
Operand1	C	S	A	N		N	P	I									yes	no
Operand2	C	S	A	N		N	P	I									yes	no
Operand3		S	A		A	N	P	I	F	B							yes	yes
Operand4		S	A		A	N	P	I	F	B							yes	yes

Related Statement: COMPUTE

Function

The DIVIDE statement is used to divide two operands.

Result Field

The result field may be a database field or a user-defined variable.

The ROUNDED clause causes the result to be rounded.

If the keyword GIVING is used, *operand2* will not be modified and the result will be stored in *operand3*; if the GIVING clause is not used, the result will be stored in *operand2*. If *operand2* is a constant or a non-modifiable Natural system variable, the GIVING clause is required.

If a database field is used as the result field, the division only results in an update to the internal value of the field as used within the program. The value for the field in the database remains unchanged.

The number of decimal positions for the result of the division is evaluated from the result field (that is, *operand2* if no GIVING clause is used, or *operand3* if the GIVING clause is used).

For the precision of the result, see also Precision of Results for Arithmetic Operations in the Natural Reference documentation.

Division by Zero

If an attempt is made to use a divisor (*operand1*) which is "0", either an error message or a result equal to "0" will be returned; this depends on the setting of the session parameter ZD (which is described in the Natural Reference documentation).

REMAINDER Option

If the keyword REMAINDER is specified, the remainder of the division will be placed into the specified field (*operand4*).

If GIVING and REMAINDER are used, none of the four operands may be an array range.

Internally, the remainder is computed as follows:

1. The quotient of the division of *operand1* into *operand2* is computed.
2. The quotient is multiplied by *operand1*.
3. The product of this multiplication is subtracted from *operand2*.
4. The result of this subtraction is assigned to *operand4*.

For each of these steps, the rules described under Precision of Results for Arithmetic Operations in the Natural Reference documentation apply.

Example

```

/* EXAMPLE 'DIVEX1': DIVIDE
/*****
DEFINE DATA LOCAL
1 #A (N7) INIT <20>
1 #B (N7)
1 #C (N3.2)
1 #D (N1)
1 #E (N1) INIT <3>
1 #F (N1)
END-DEFINE
/*****

DIVIDE 5 INTO #A
WRITE NOTITLE 'DIVIDE 5 INTO #A' 20X '=' #A
/*****
RESET INITIAL #A
DIVIDE 5 INTO #A GIVING #B
WRITE 'DIVIDE 5 INTO #A GIVING #B' 10X '=' #B
/*****
DIVIDE 3 INTO 3.10 GIVING #C
WRITE 'DIVIDE 3 INTO 3.10 GIVING #C' 8X '=' #C
/*****
DIVIDE 3 INTO 3.1 GIVING #D
WRITE 'DIVIDE 3 INTO 3.1 GIVING #D' 9X '=' #D
/*****
DIVIDE 2 INTO #E REMAINDER #F
WRITE 'DIVIDE 2 INTO #E REMAINDER #F' 7X '=' #E '=' #F
/*****
END

```

DIVIDE 5 INTO #A	#A:	4
DIVIDE 5 INTO #A GIVING #B	#B:	4
DIVIDE 3 INTO 3.10 GIVING #C	#C:	1.03
DIVIDE 3 INTO 3.1 GIVING #D	#D:	1
DIVIDE 2 INTO #E REMAINDER #F	#E:	1 #F: 1